



# PREMIUM AUDIT

NAME OF PROJECT	When Lambo Moon Binance
TOKEN BEP20 ADDRESS	0xABFe46a1ffF73608431A1Ce82ebdBB96785FDbA6
WEBSITE / TELEGRAM	<a href="https://whenlambomoonbinance.com/">https://whenlambomoonbinance.com/</a>



**DISCLAIMER: PLEASE READ FULL AUDIT**



# IMPORTANT DISCLAIMER

---

I.

## **NOT RESPONSIBLE**

Our audit report is not responsible for anything if the project audited turn into a scam.

II.

## **NOT GUARANTEE**

Analytix Audit in no way guarantees that a project will not remove liquidity, sell off team tokens, or exit scam.

III.

## **INFORMATION**

Analytix Audit do a research and provides public information about the project in an easy-to-understand format for the common person.

IV.

## **AUDIT AGREE**

Agreeing to an audit in no way guarantees that a team will not remove all liquidity ("Rug Pull"), remove liquidity slowly, sell off tokens, quit the project, or completely exit scam

V.

## **ENCOURAGE INVESTMENT**

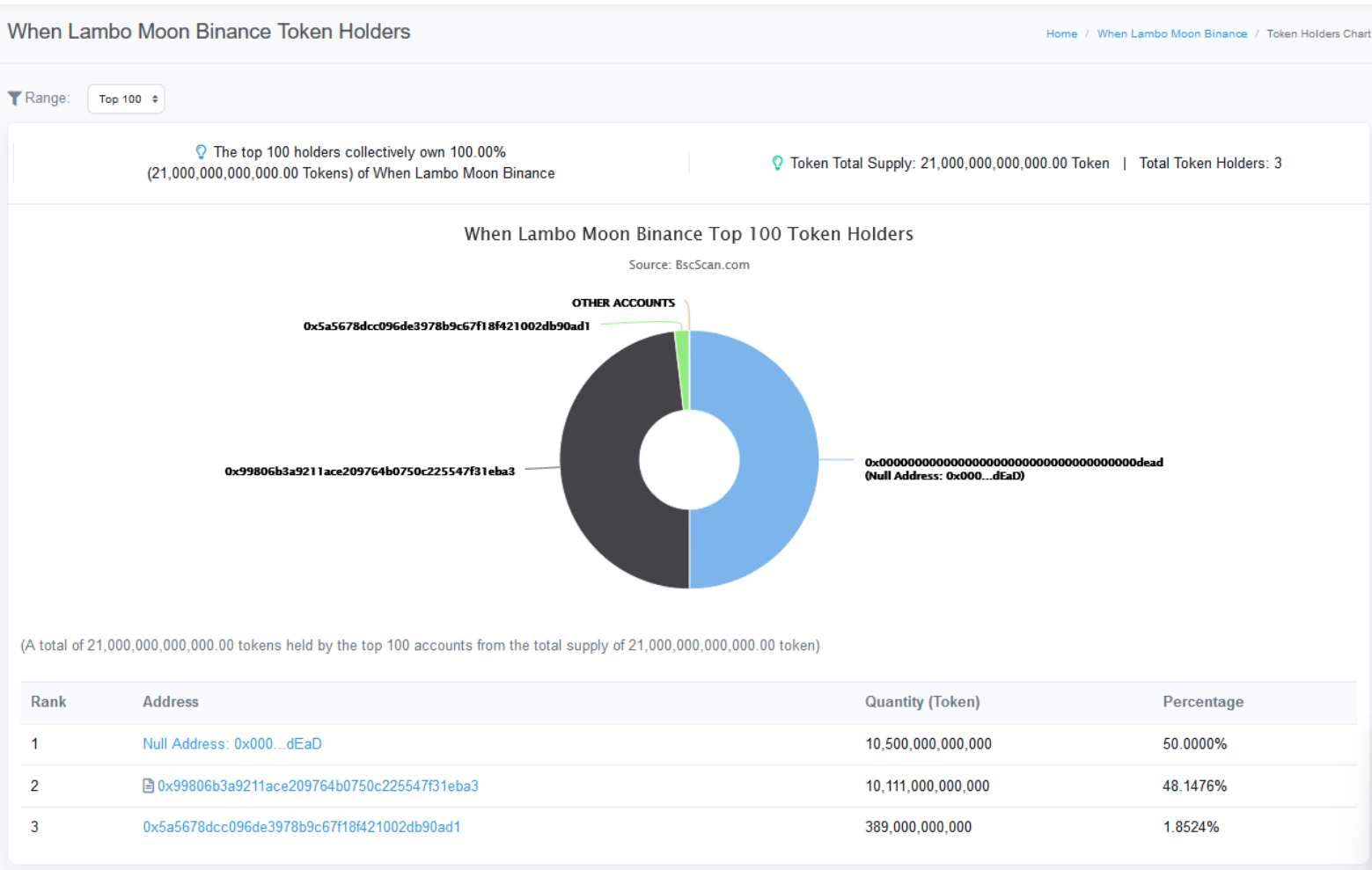
Analytix Audit in no way takes responsibility for any losses, nor does Analytix Audit encourage any speculative investments

VI.

## **INFORMATION**

The information provided in this audit is for information purposes only and should not be considered investment advice. Analytix Audit does not endorse, recommend, support, or suggest any projects that have been audited.

# TOKENOMICS - OVERVIEW



(This comment refers to when the audit was carried out maybe team locked tokens later)

# CONTRACT CODE - OVERVIEW



## QUICK REVIEW

- Fees:
  - Buy: 16%
  - Sell: 16%
- Owner can change fees up to 25%
- Owner can set Tx Limits

# Authorized Privileges:

```
520 * Requirements:
521 *
522 * - the calling contract must have an ETH balance of at least `value`.
523 * - the called Solidity function must be `payable`.
524 *
525 * _Available since v3.1._
526 */
527 function functionCallWithValue(
528     address target,
529     bytes memory data,
530     uint256 value
531 ) internal returns (bytes memory) {
532     return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
533 }
534
535 /**
536 * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256}[`functionCallWithValue`], but
537 * with `errorMessage` as a fallback revert reason when `target` reverts.
538 *
539 * _Available since v3.1._
540 */
541 function functionCallWithValue(
542     address target,
543     bytes memory data,
544     uint256 value,
545     string memory errorMessage
546 ) internal returns (bytes memory) {
547     require(address(this).balance >= value, "Address: insufficient balance for call");
548     require(isContract(target), "Address: call to non-contract");
549
550     (bool success, bytes memory returndata) = target.call(value: value)(data);
```

```
modifier authorized() {
    function MetaBurn(uint256 amount) external authorized {
function setAutoBuybackSettings(bool _enabled, uint256 _cap, uint256 _amount, uint256 _period)
    external authorized {
    function launch() public authorized {
    function setTxLimit(uint256 amount) external authorized {
    function setBlacklisted(address account, bool value) external authorized {
    function setIsDividendExempt(address holder, bool exempt) external authorized {
    function startTrading() external authorized {
    function burnTokens(uint256 amount) external authorized {
function TransferBNBsOutfromContract(uint256 amount, address payable receiver) external
    authorized {
    function setIsFeeExempt(address holder, bool exempt) external authorized {
    function setIsTxLimitExempt(address holder, bool exempt) external authorized {
function setFees(uint256 _liquidityFee, uint256 _buybackFee, uint256 _reflectionFee, uint256
    _marketingFee, uint256 _feeDenominator) external authorized {
function setFeeReceivers(address _autoLiquidityReceiver, address _marketingFeeReceiver) external
    authorized {
    function setSwapBackSettings(bool _enabled, uint256 _amount) external authorized {
    function setTargetLiquidity(uint256 _target, uint256 _denominator) external authorized {
function setDistributionCriteria(uint256 _minPeriod, uint256 _minDistribution) external authorized {
    function setDistributorSettings(uint256 gas) external authorized {
```

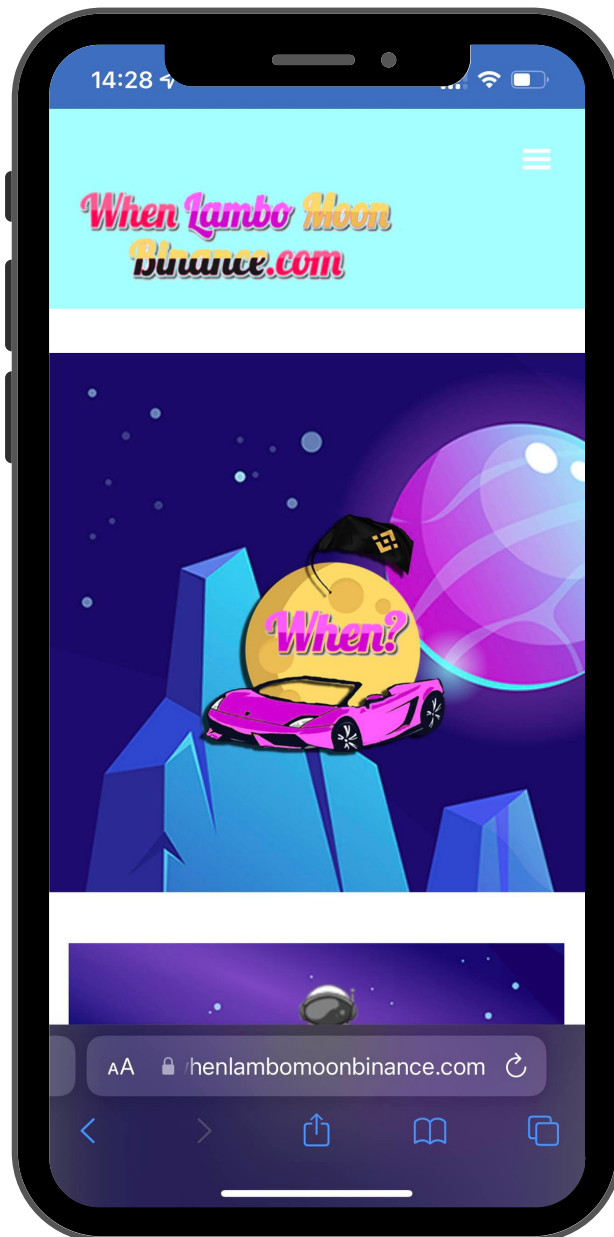
# Issue Checking:

N°	Issue description.	Checking Status.
1	Compiler Errors	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module	Passed
21	Fallback function security.	Passed

Audit Result:

**PASSED**

# WEBSITE - OVERVIEW



🕒 **Domain Registration: 2022-01-30**

38 days old

**SSL CERTIFICATE: A+**

(<https://www.ssllabs.com>)

# **SOCIAL MEDIA - OVERVIEW**



**WHENLMB**



**LAMBOBINANCE**



**WHEN\_LAMBO\_MOON\_BINANCE**



**WHENLAMBOMOONB**